

Problem B. Выставка

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Известный художник проводит выставку своих картин в местной галерее. Художник готов выставить на обозрение k работ, однако в галерее есть всего лишь n держателей для картин. i -ый держатель может держать не более одной картины весом не более d_i грамм. Для каждой картины известна ее эстетическая ценность a_i и вес w_i в граммах. Помогите художнику выбрать картины для выставки таким образом, чтобы их можно было безопасно укрепить на держателях, при этом суммарная эстетическая ценность была максимальна.

Input

В первой строке записано два целых числа n и k ($1 \leq n \leq k \leq 10^4$).

Во второй строке записано n целых чисел d_1, \dots, d_n — максимальные веса каждого из держателей ($1 \leq d_i \leq 10^6$).

Следующие k строк содержат по два целых числа a_j и w_j каждый — эстетическая ценность и вес каждой из картин ($1 \leq a_j, w_j \leq 10^6$).

Output

Выведите n чисел, разделенных пробелами — номера картин для выставки. i -ое число должно равняться номеру картины, которую необходимо повесить на i -ом держателе, либо 0, если держатель нужно оставить пустым.

Example

| standard input | standard output |
|---|-----------------|
| 5 10 1 2 3 4 5 10 3 4 3 11 8 1 5 5 8 7 1 5 5 8 3 4 2 7 3 | 6 9 1 8 10 |

Problem C. Игра

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 512 mebibytes

Петя и Вова играют в игру. На игровом поле отмечено n полян, пронумерованных от 1 до n . Некоторые поляны соединены стрелочками, при этом из каждой поляны исходит хотя бы одна стрелочка. В начале игры Петя ставит фишку на какую-то поляну. Вова знает конфигурацию поля, но не знает, какую поляну выбрал Петя.

На своем ходу Вова называет номер поляны k . Если Петина фишка находится на поляне k , Петя говорит «убит» и игра заканчивается. Иначе Петя говорит «мимо», после чего перемещает фишку по любой стрелочке, исходящей из его текущей поляны (также в тайне от Вовы).

Ваша задача — определить минимальное количество ходов, за которое Вова может заведомо закончить игру, не имея информации об изначальном положении и перемещениях Петиной фишки.

Input

В первой строке записаны числа n и m ($1 < n \leq 20$, $0 \leq m \leq n(n-1)$) — число полян и стрелок на поле соответственно.

Следующие m строк описывают стрелки. Каждая из этих строк содержит два различных числа — номер стартовой и конечной поляны для очередной стрелки.

Output

Если Вова не может заведомо завершить игру, выведите «No solution». Иначе, в первой строке выведите число k — минимальное число ходов в Вовиной стратегии. В следующей строке выведите k чисел — номера полян, которые должен называть Вова.

Example

| standard input | standard output |
|----------------|-----------------|
| 3 3 | 3 |
| 1 2 | 1 1 1 |
| 2 3 | |
| 3 1 | |

Problem D. Ханойские башни

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Дано три стержня A, B, C. Изначально на стержне A расположено n дисков попарно различного диаметра таким образом, что самый маленький диск расположен на самом верху, следующий расположен выше всех остальных, и т.д. Требуется переместить все диски со стержня A на стержень B, используя стержень C как вспомогательный. За один ход можно переместить верхний диск с любого стержня на любой другой, если новый стержень пуст либо диск оказывается на другом диске большего диаметра.

Классическое решение этой задачи описывается следующей функцией:

```
void Hanoi (int a, int b, int c, int n) {  
    if (n>0) {  
        Hanoi (a, c, b, n - 1);  
        cout << "disc " << n << " from " << a " to " << b << endl;  
        Hanoi (c, b, a, n-1);  
    }  
    return;  
}
```

| Step | Disk | From | To | Combination |
|------|------|------|----|-------------|
| 0 | | | | AAA |
| 1 | 1 | A | B | BAA |
| 2 | 2 | A | C | BCA |
| 3 | 1 | B | C | CCA |
| 4 | 3 | A | B | CCB |
| 5 | 1 | C | A | ACB |
| 6 | 2 | C | B | ABV |
| 7 | 1 | A | B | BBV |

Хорошо известно, что это решение требует $2^n - 1$ перекладываний. С учетом стартовой конфигурации, в процессе решения получается 2^n различных расположений дисков по стержням. Отсюда легко видеть, что не все возможные распределения дисков встречаются в процессе решения, например, при $n = 3$ комбинация «СAB» не встречается (в этой комбинации самый маленький диск находится на стержне C, средний на стержне A и большой на стержне B).

Ваша задача в том, чтобы определить по заданной конфигурации, встречается ли она в процессе игры.

Input

В первой строке записано одно целое число n — число дисков ($1 \leq n \leq 250$). Во второй строке записано n заглавных букв ('A', 'B' or 'C') — расположение n дисков на стержнях. Первый символ описывает стержень, на котором находится самый маленький диск, второй — стержень, на котором находится следующий по величине диск, и т.д.

Output

Выведите «YES» либо «NO» в зависимости от того, встретится ли данное распределение в игре.

Examples

| standard input | standard output |
|----------------|-----------------|
| 3 ACB | YES |
| 3 CAB | NO |

Problem E. Робот

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Для экспедиции по исследованию спутников Юпитера байтландские робототехники разработали специального робота. Сейчас проходят испытания. Одно из испытаний выглядит следующим образом.

В верхнем левом углу поля $M \times M$ стоит робот. Некоторые клетки поля содержат ловушки, попав в которые, робот не сможет продолжать движение. Робот понимает два типа команд: 0 — робот идёт на одно поле вправо, 1 — робот идёт на одно поле вниз. При этом память робота вмещает не более k команд, то есть программа робота представляет собой строку из нулей и единиц длины не большей, чем k .

Команды выполняются по очереди; после завершения последней команды выполняется первая (вне зависимости от того, какова длина программы, то есть программа 01 задаёт движение робота «вправо-вниз» по циклу).

Требуется построить программу для робота, которая выведет его за границы поля. Если таких программ несколько, требуется построить программу, состоящую из наименьшего количества команд. Если и таких программ несколько, требуется выбрать лексикографически минимальную из них.

Input

В первой строке входа заданы два целых числа M и k — количество полей на одной вертикали (или горизонтали) и максимальная длина программы робота ($1 \leq M \leq 1000$, $1 \leq k \leq 50$). В последующих M строках заданы по M символов, описывающих поле. В левом верхнем углу стоит робот (обозначаемый буквой 'R'). Поля, содержащие ловушки, обозначены символом 'X', проходимые поля — символом '.'. Гарантируется, что хотя бы одна программа длины не более k , выводящая робота за границы поля, существует.

Output

Выведите строку длиной не более k , состоящую из нулей и единиц — лексикографически наименьшую из кратчайших программ, выводящих робота за границы поля.

Example

| standard input | standard output |
|---|-----------------|
| 6 5 R.X... X...X. XXX... .X.XX. X.X.XX XXX... | 010 |

Problem F. Сортировка по сумме цифр

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Выпишем числа от 1 до n , и упорядочим их в порядке возрастания суммы цифр в десятичной записи (при равенстве суммы цифр числа идут по возрастанию). Например, при $n = 20$ получается следующая последовательность: 1, 10, 2, 11, 20, 3, 12, 4, 13, 5, 14, 6, 15, 7, 16, 8, 17, 9, 18, 19.

Определите позицию числа k в полученной последовательности.

Input

Первая строка ввода содержит два целых числа n и k ($1 \leq k \leq n \leq 10^{12}$).

Output

Выведите одно число — позицию числа k .

Example

| standard input | standard output |
|----------------|-----------------|
| 20 13 | 9 |

Problem G. Тарифные зоны

Input file: *standard input*
Output file: *standard output*
Time limit: 5 seconds
Memory limit: 512 mebibytes

Пригородное железнодорожное сообщение в окрестностях столицы Байтландии устроено следующим образом: есть n станций, соединённых n прямыми двухпутными ветками так, что каждые две станции соединены напрямую не более, чем одной веткой и между любыми двумя станциями существует железнодорожное сообщение.

Управление железной дороги хочет разбить n станций на тарифные зоны следующим образом:

- в каждой зоне количество станций одинаково;
- между любыми двумя станциями, расположенными в одной зоне, можно проехать или напрямую, или только через станции, расположенные в той же самой зоне.

Требуется найти количество способов это сделать.

Input

Вход состоит из нескольких тестовых примеров.

Первая строка каждого тестового примера содержит одно целое число n ($3 \leq n \leq 10^5$) — количество станций. Вторая строка содержит n целых чисел; i -е из этих чисел a_i обозначает, что станции с номерами i и a_i соединены прямой веткой ($1 \leq a_i \leq n$, $a_i \neq i$, $a_{a_i} \neq i$).

Гарантируется, что сумма всех n во входном файле меньше 10^6 .

Output

Для каждого тестового примера выведите одно целое число — требуемое количество способов.

Example

| standard input | standard output |
|----------------|-----------------|
| 6 | 4 |
| 3 4 5 6 1 2 | 3 |
| 6 | |
| 4 5 4 2 1 2 | |